REMOTELY CONTROLLED

SELF-BALANCING

MINIBIKE

Team 22

Member: Jiaming Xu(jx30), Will Chen(hongyuc5), Eric Tang(leweit2)

Spring 2024

TA: Jason Zhang

ECE445

1. Introduction

1.1 Problem

Bike share and scooter share programs have surged in popularity across the globe in recent years. This mode of transportation is increasingly being recognized and supported for its convenience and environmental benefits. In Champaign, a company named Veo provides such services, offering a convenient option for short-distance travel between school buildings and bus stops. However, a notable problem is preventing more people from using this kind of transportation; these bikes can be parked randomly throughout the city. Inevitably, we will have bikes unevenly distributed around an area. The result is users often need to search for and walk to the nearest bike location, which can detract from the convenience of the service.

Several potential solutions to this issue have been considered, but each has its drawbacks. For instance, periodically collecting and redistributing all of the bikes could address the problem of uneven bike distribution. However, this approach is likely to be costly and inefficient, requiring significant logistical effort and resources. Another potential solution involves flooding the region with a high number of bikes to ensure availability in most areas. Yet, this method raises concerns about cost efficiency and potentially cluttering public spaces with excessive bikes and scooters.

1.2 Solutions

To solve this problem practically, we need tracking and management systems for bikes and scooters, that encourage users to redistribute bikes. But more importantly, use the tracking and management systems to deliver the bikes to users. To achieve this, we first need to have a self-driving bike or scooter. This is the goal of our project. First, we will construct a 40 cm-long bike model, incorporating a gyroscope, two 11V motors, one 3.3V motor, a flywheel, a control system, and an 11V battery. Second, we will utilize data from the gyroscope to adjust the torque on the flywheel, ensuring the bike remains balanced. Third, we will use one of the 11V motors to propel the rear wheel, enabling the bike to drive straight, while the 3.3V motor will be tasked

with controlling the direction. Finally, we will connect the bike's controller to an external controller, enabling the bike's remote operation.
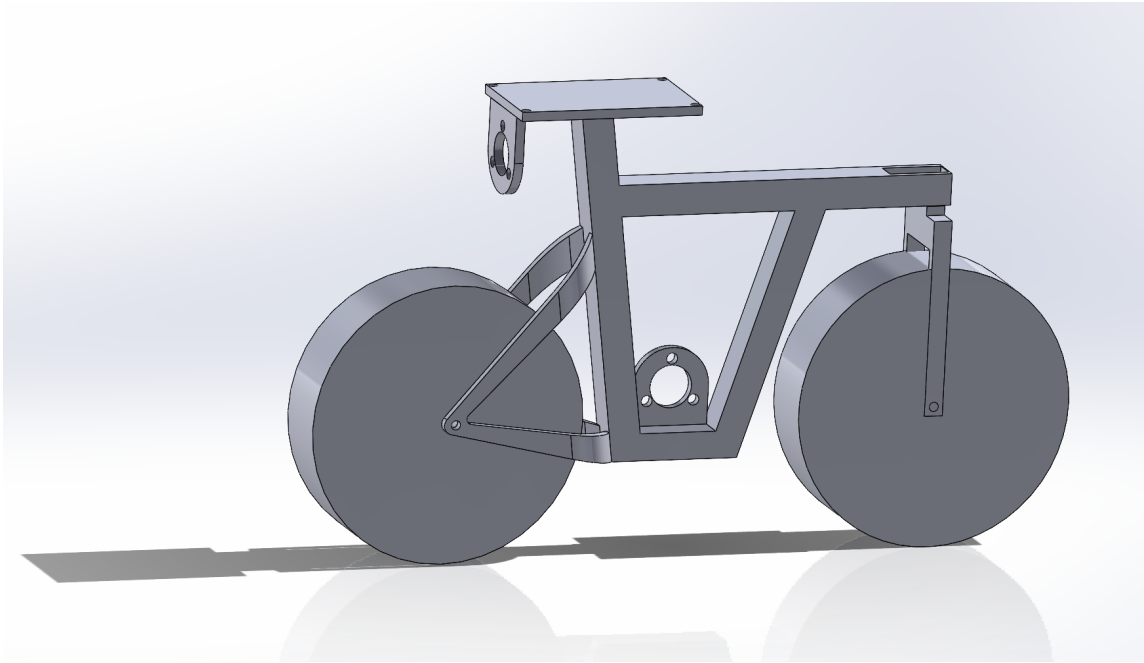
1.3 Visual Aid

Figure 1: Visual Aid for bike Frame

1.4 High-Level Requirements

The bicycle should be able to remain self-balanced when the power is on.

The bicycle should be able to self-regulate lateral force disturbances within +- 10 degrees and continue to maintain balance.

The bicycle should be remotely controlled by the controller within a radius of 5 m from a bicycle and can maintain self-balancing during controlled movement.
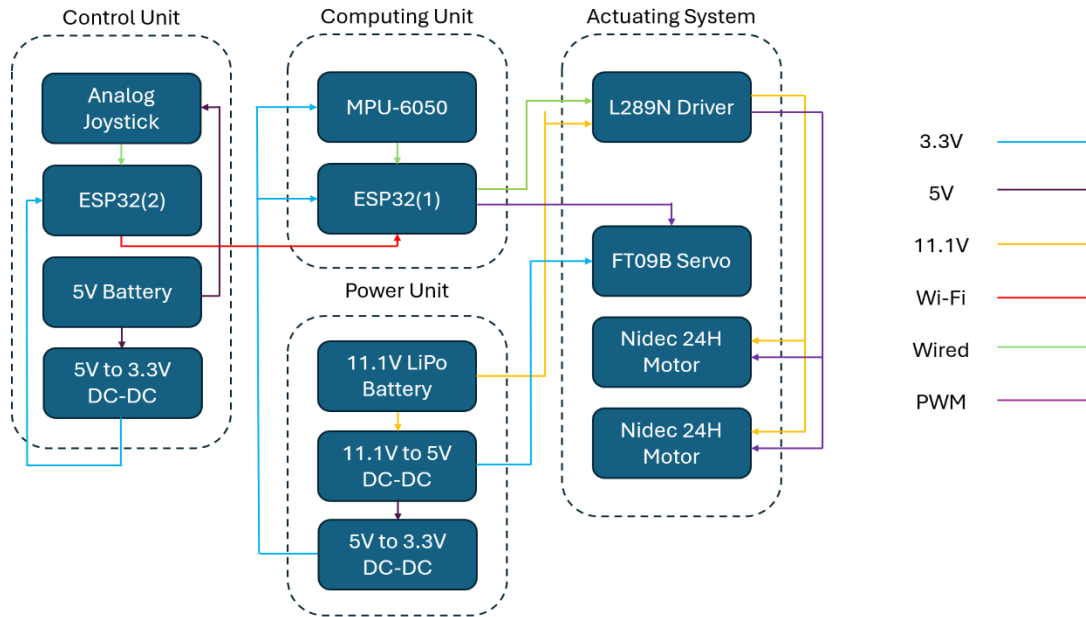
2. Design

2.1 Block Diagram

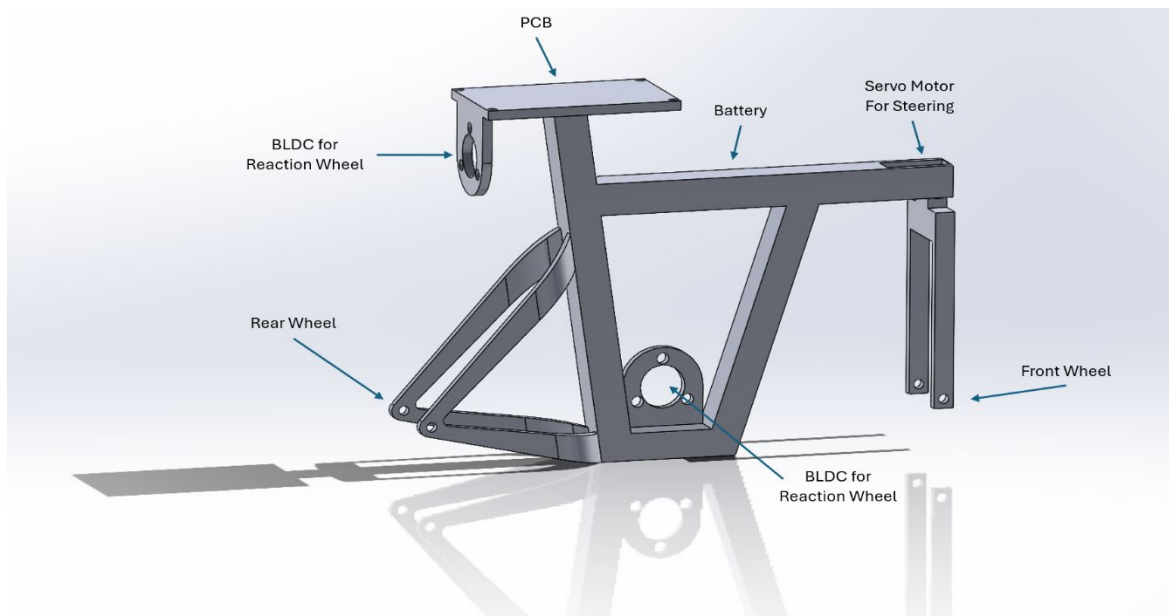Figure 2: Block Diagram

2.2 Physical Design

Figure 3: Physical Design

We designed the minibike frame with SolidWorks, and we plan to print the frame out with PETG filament and a 3D printer. All electronics and two wheels will be installed to predefined positions on the bike frame with screws and zip ties.

2.3 Subsystems

2.3.1 Control Subsystem

We plan to design a Wi-Fi 2.4GHz remote controller using two ESP32-S3-WROOM microcontrollers and a two-axis pushbutton rocker sensor module by joystick potentiometer to implement the function of remote control. One ESP32-S3-WROOM serves as the transmitter which functions as the primary input source, providing control inputs such as acceleration, braking, and steering. It communicates these inputs wirelessly to another ESP32-S3-WROOM microcontroller function as a receiver. The ESP32-S3-WROOM Microcontroller decodes and processes these data packets to extract control commands, including steering angle, brake intensity, and any additional functions mapped to the controller buttons. The communication between two ESP32 microcontrollers will be coded based on ESP_NOW, which is a wireless peer-to-peer Wi-Fi communication protocol. ESP-NOW utilizes Wi-Fi technology, which typically offers better range and stability compared to Bluetooth or traditional RF protocols. It also supports higher data rates compared to Bluetooth and RF protocols, allowing for faster transmission of control commands and data between the remote controller and the minibike, which is beneficial for our project since it requires real-time control.

For the transmitter part, we designed the pin connection in the following:

Joystick Pins	ESP32 DevBoard Pins
GND	GND
5V	3.3V
VRX	IO32
VRY	IO33
SW	IO25

Requirement	Verification
R1. The remote control must maintain stable communication with the bicycle with a delay <=20ms	V1. We can add a timestamp when the microprocessor receives the data from the remote controller. The time delay is the time difference between time stamps. We can let the remote control send the signals to the microprocessor continuously. By tracking the time intervals for each signal sending, we can check whether or not the delay is <= 20ms
R2. The control signal receiver can receive the signal from the transmitter successfully	V2. We will add print statements on the microcontroller's script to check whether the signal that the transmitter is sending out is the same as the signal the receiver receives. The

	print statements can be observed on the computer screen directly
R3. The remote controller should be able to control the minibike steadily within a range of 50 meters	V3. We first measured the distance of 50m with a soft ruler, and then moved away from the bicycle in units of 10m with the bicycle as the center of the circle and used the remote controller to send forward, backward, and steering commands to observe whether it was successful and delayed within the acceptance range

2.3.2 Actuating Subsystem

The Actuating Subsystem consists of two 11V motors and a 5V servo motor. One of the 11V electric motors is used for propelling the bike by utilizing a chain/belt system that connects to the rear wheel. The other 11V motor is used for powering the flywheel to input a new torque into the bike system and balance the bike when it is leaning. The 5V servo is used to steer the front wheel. The motor control needs to control the high-power motor to do quick input changes. It needs to change the torque exerted by one electric motor onto the flywheel in a very short amount of time.

Requirements	Verification
R1. The Flywheel needs to be balanced to spin at a high rpm	V1. Attach the motor with the flywheel and start to spin the flywheel when attached to a stick. Watch the flywheel and motor's vibration, if the stick shakes badly we need to rebalance the flywheel, or we have to lower the rpm for more stable balancing components.

R2. The Flywheel motor is strong and precise enough to mitigate torque	V2. Attach the motor with the flywheel on a torque measuring stick and let the motor spin the flywheel while securing the stick. When the flywheel reaches a desired rpm, release the stick and test changing the rpm's influence on the stick. Test when the stick is tilted can the motor change torque quickly enough to rebalance the stick. When the assembly passes this test, it needs to be tested when implemented on the bike with wheels and other components installed.
R3. The Propelling motor is strong enough to power the bike, and the chain/belt is strong enough to hold the torque	V3. Implement the system onto the bike and test it with components and weights on the bike. If it can propel the bike on a 20-degree slope.
R4. The steering servo is quick and precise	V4. Test the lag of servo input and test the torque it can offer.
R5. The software logistic controls 2 11V motors properly based on tilted angle	V5. Test the subsystem before implementing the subsystem onto the bike model. Test if the Motor Control System alters the inputs of two motors. Test if the motor can withstand sudden changes when the flywheel is attached, and test if we can use the controller to control this subsystem.
R6. The motor can output a torque that is strong and quick enough to balance the bike model	V6. Implement the above design on our bike model with the battery and other two motors installed and test if the motor control system

	is quick and precise enough to correct tilts

2.3.3 Computing Subsystem

The Computing Subsystem consists of an MPU6050 accelerometer and an ESP32 S3 WROOM microcontroller. The MPU6050 will take measurements and feed that information into the microcontroller; then, the microcontroller will process this information and calculate the tilt angle of the bike and appropriate input to the motor that connects to the reaction wheel.
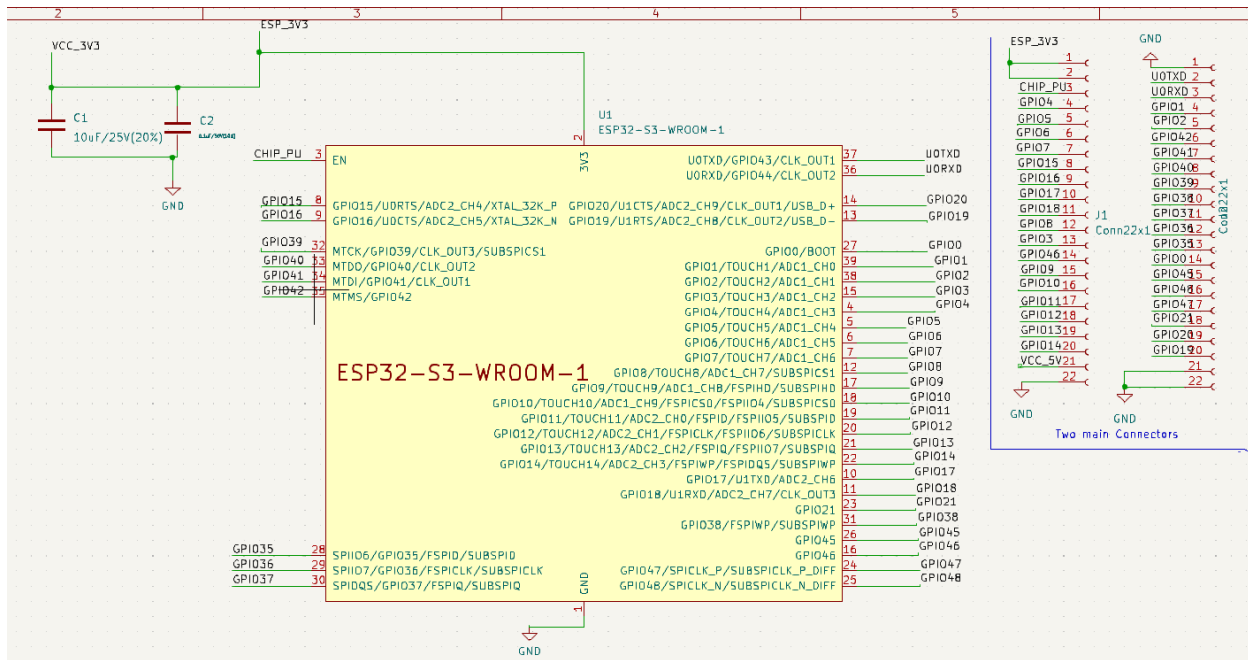

Figure 4: ESP32-S3-WROOM-1 Pin Assignment

Requirements	Verification
R1. The microcontroller can interpret the correct tilt angle based on measurement.	V1. Test with several given angles without attaching to the bike. Solely testing the accuracy of the accelerometer (accelerometer might be complicated, so, we will test the accelerometer together with the software that

	calculates the tilt angle.)
R2. The microcontroller can interpret the correct tilt angle based on measurement in moving situations	V2. Test with several given angles after being attached to the bike and have the bike running in several different road conditions.

2.3.4 Power Subsystem

The power subsystem of our project serves as the backbone, ensuring stable and reliable operation of all components. At its core is an 11.1V LiPo battery, strategically chosen for its capacity and voltage output. This battery not only powers the entirety of our project but also directly supplies energy to drive DC motors, essential for the propulsion of our mini car. To cater to the diverse power requirements of various components, we've implemented two voltage regulators. The first, an 11.1V to 5V buck regulator, efficiently steps down the battery voltage to a stable 5V output. We plan to use LM2596 for this regulator. This regulated 5V supply is dedicated to powering the FT09B Servo motor, ensuring consistent torque delivery for precise control. The second regulator, a 5V to 3V linear voltage regulator, further refines the power output, providing a clean and stable voltage source for delicate components such as the ESP32s and MPU6050. With this meticulously designed power subsystem, we ensure optimal performance, longevity, and reliability across all facets of our mini car project.
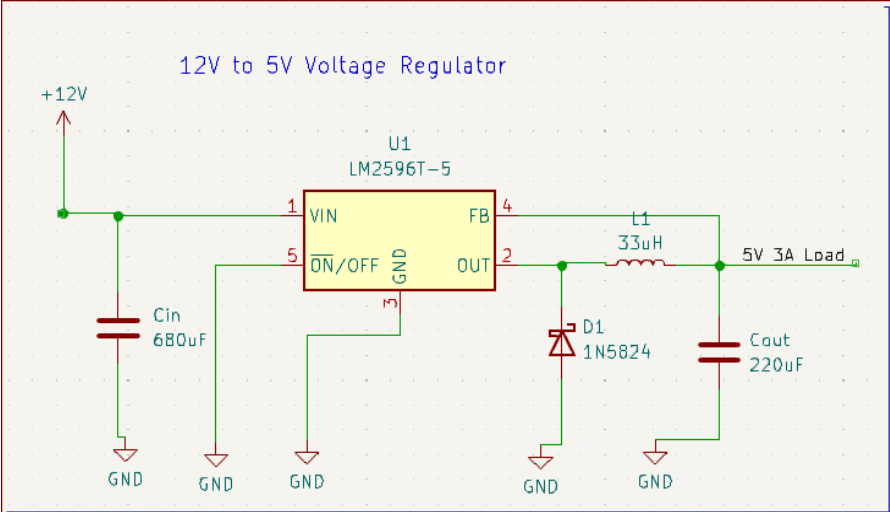
Figure 5: 12V to 5V Voltage Regulator Pin Assignment

Requirements	Verification
R1. The 11.1V to 5V buck voltage regulator should output stable 5V	V1. Utilize a multimeter to gauge the input and output voltage and current, verifying they fall within predefined thresholds.
R2. The 5V to 3.3V linear voltage regulator should output stable 3.3V	V2. Utilize a multimeter to gauge the input and output voltage and current, verifying they fall within predefined thresholds.
R3. The 11.1V LiPo battery should supply the whole project at a 3A >= 30min	V3. Use a timer to ensure it lasts for at least 30 minutes load. Utilize a multimeter to gauge the input and output voltage and current, verifying they fall within predefined thresholds.

2.4 Voltage Management and Operating Time

Our battery is an 11.1V 1500mAh Li-Po battery. We can convert it to Wh which is:

$$11.1V * 1.5Ah = 16.65 \text{ Wh}$$

Part	Average Current Draw @ 5 V	Total Current: 0.418A
Processor	150mA	
Motor Controller	50mA	

Servo Motor	Average of 300mA when active, and 10mA when idle, we assume the Servo needs to turn the wheel around 70% of the time thus the average current draw is: $$300\text{mA} * 70\% + 10\text{mA} * 30\% = 213\text{mA}$$	To be safe we give an extra 20% of the designed current needed. Thus, we will design with a 0.5A current draw from 5V system
accelerometer	5mA	

For our 5v system with an efficiency(eff) of 85%, thus it will have an average power of

$$P_{avg} = V_{out} * I_{avg} / eff = 5V * 0.5A / 85\% = 2.95W$$

For the two 11V electric motors, the propelling wheel and flywheel should both have a relatively stable operating environment, thus I assume in general 15% of the time running at maximum power and 85% running under normal operating current(0.7A). This means

$$P_{avg} = V_{11} * I_{avg} = 11V * 2 * (0.7A * 85\% + 1.8A * 15\%) = 19W.$$

Thus, for normal operation the system has a net power consumption of 21.95W, which means under normal operating circumstances the bike can run 45min on a single battery.

For extreme cases, we assume that the propelling wheel is running at maximum torque, and the flywheel motor runs at maximum torque for 50% of the time.

$$P_{avg} = V_{11} * I_{avg} = 11V * ((0.7A * 50\% + 1.8A * 50\%)+(1.8*100\%)) = 33.55W$$

Thus, our total power consumption of the system is 36.5W, which means the system can run on a single battery for 30 minutes.

2.8 Tolerance Analysis

Our project has two 11V motors connected parallel to one 11V power source. A voltage regulator(LM2596)[5] converts to 5 V, which powers all control units and steering servo motors.

2.8.1 Parts Current

Part	Worst Case Current Draw @ 5 V	Total Current:

		1.005A
Processor	250mA	
Motor Controller	50mA	
		To be safe we give an extra 20% of the designed current needed. Thus, we will design with 1.2A
Servo Motor	700mA	
accelerometer	5mA	

2.8.2 Variables

Variable	Value
Maximum temperature(LM2596)	125 C
Efficiency(eff) when operating @ 5V	85%
Voltage output(V_{out})	5V
Junction-to-case resistance(Θ_{jc})	2 C/W
Thermal resistance(Θ_{ca})	30 C/W
Environment temperature(T_{env})	38 C

2.8.3 Calculation of heat tolerance

First we calculate the power output of the system:

$$P_{out} = V_{out} * I_{in} = 5V * 1.2A = 6W$$

Then we use the efficiency of voltage regulator to calculate the power dissipated, and assume all power dissipated into the generation of heat

$$P_{diss} = P_{in} - P_{out} = P_{out}/eff - P_{out} = 1.06W$$

Now we can use the power that generates heat and the environment temperature to calculate a theoretical value for operating temperature:

$$T_{op} = P_{diss} * (\Theta_{jc} + \Theta_{ca}) + T_{env} = 1.06 *(2 + 30) + 38 = 71.92 \ C \ < 125 \ C(T_{max})$$

The operating temperature is well below the maximum temperature of LM2596. Thus we can conclude that our system will operate under the limit of LM2596, we have around 70% tolerance of the original design.

3. Cost and Schedule

3.1 Cost Analysis

3.1.1 Labor:

We assume that the average salary of an ECE graduate is about $50 per hour. This project is composed of 3 members, and we contribute roughly 14 hours per week. We estimate that this project will take about 10 weeks to complete.

Thus, the total labor cost = $50 * 3 * 14 * 10 = $21,000.

3.1.2 Parts

Item Name	Manufacturer	Quantity	Unit Price	Cost	Description
LM2596	Texas Instrument	1	$4.50	$4.50	11V to 5V DC to DC
L298N	DFRobot	2	$12.9	$25.8	Motor Driver
SG90	Beffkkip	1	$7.99	$7.99	Servo Motor
MPU 6050	Foriot	1	$8.69	$8.69	Accelerometer
Wheel	Uxcell	2	$8.00	$16.00	Wheels
LiPo 11.1V	Ovonic	1	$16.50	$16.50	Battery
PETG Filament	Overture	1	$16.14	$16.14	Filament for 3D Print
Joystick	HiLetgo	1	$6.29	$6.29	Joystick control
BLDC 24H	Nidec	2	$5.90	$11.80	BLDC motor
Circuit elements	N/A	N/A	$15.00	$15.00	Common PCB circuit elements
ESP32-S3-WROOM	HiLetgo	2	$9.99	$19.98	Microcontroller

3.1.3 Grand total costs

Our estimated labor costs will be $21,000, and our estimated cost of all parts will be $148.69. Thus, our total cost will be $21,148.69.

3.2 Schedule

Week	Task	Team Member
Week-07	Design Review Bike Frame prototyping PCB Design	All Jiaming All
Week-08	PCB Design Motor Testing Remote Controller	Eric Jiaming Will
Week-09	N/A (Spring Break)	N/A
Week-10	PCB Refining Self-balancing tunning Remote Controller Testing	Eric Jiaming Will
Week-11	PCB Refining Bike Movement	Eric Jiaming & Will
Week-12	Test and refine	All
Week-13	Final Testing Mock Demo	All All
Week-14	Final Demo and mock presentation	All
Week-15	Final Presentation	All

4. Discussion of Ethics and Safety

Throughout this project, we adhere to the IEEE Code of Ethics. We will "uphold the highest standards of integrity, responsible behavior, and ethical conduct in professional activities"[1]. We will "treat all persons fairly and with respect, to not engage in harassment or discrimination, and to avoid injuring others"[1]. We will "strive to ensure this code is upheld by colleagues and co-workers"[1].

We recognize that there are some potential safety concerns during this project. The workstation must be kept clean and organized, and tools such as soldering irons must be shut down immediately after use. We will clean up the workstation right after use take a picture upload it to Google Drive and annotate the date.

When handling motors, we must ensure that we have a clean, spacious, space area. We need to check the number of people around, potential electrical dangers such as ponds or wires, and the cleanliness of the surface the cart is going to be running. Safety is always the top priority throughout this project, and every team member needs to follow the procedure above.

The Lithium battery must be kept safely inside a fire-retardant charging bag for storage and charging. For storage a few days long, we need to find a dry and cool place that won't be directly hit by sunlight. For storage longer than a few days, we need to follow all the instructions above and use a meter to measure the voltage and make sure their voltage is around 4.4V. When charging the battery, we can only use the provided charger from the manufacturer. During usage, we will charge the 11V battery frequently to ensure the power of our 11V motors.[2]We will do voltage testing every time after charging the battery and document the voltage, and visual inspection every time before and after charging the battery. When batteries are not in use we need to disconnect them from the cart and store it in a dry enclosed transparent box. If the battery shows any signs of leakage or malfunction, we must immediately stop using it and depose it appropriately.

5. Citations

1.   "IEEE Code of Ethics." *IEEE*, https://www.ieee.org/about/corporate/governance/p7-8.html

2.   "Battery Safety" UIUC Division of Research Safety,
     https://drs.illinois.edu/Page/SafetyLibrary/BatterySafety